

ANISOTROPIC MESH QUALITY MEASURES AND ADAPTATION FOR POLYGONAL MESHES*

WEIZHANG HUANG[†] AND YANQIU WANG[‡]

Abstract. Anisotropic mesh quality measures and anisotropic mesh adaptation are studied for polygonal meshes. Three sets of alignment and equidistribution measures are developed, one based on least squares fitting, one based on generalized barycentric mapping, and the other based on singular value decomposition of edge matrices. Numerical tests show that all three sets of mesh quality measures provide good measurements for the quality of polygonal meshes under given metrics. Based on one of these sets of quality measures and using a moving mesh partial differential equation, an anisotropic adaptive polygonal mesh method is constructed for the numerical solution of second order elliptic equations. Numerical examples are presented to demonstrate the effectiveness of the method.

Key words. anisotropic, adaptive mesh, polygonal mesh, generalized barycentric coordinates.

AMS subject classifications. 65M60, 65M60.

1. Introduction. During the last decade polygonal/polyhedral meshes have gained considerable attention in the scientific computing community partially due to their flexibility to deal with complicated geometry, curved boundaries, and local mesh refining and coarsening and also to their connection with the Voronoi diagram. Various methods have been developed for the numerical solution of partial differential equations (PDEs) on polygonal/polyhedral meshes, including the mimetic finite difference method (see the recent survey paper [30]), conforming finite element methods using generalized barycentric coordinates [33, 40, 41, 43, 44, 46, 48], the finite volume method [19, 42], the virtual element method (see [45] and references therein), the discontinuous Galerkin method [8, 17, 35], and the weak Galerkin method [36, 47]. For the error analysis of these methods, shape regularity of polygonal/polyhedral meshes has also been studied and a number of shape regularity conditions have been proposed [4, 18, 19, 35, 47]. The minimal requirement of these shape regularity conditions unanimously states that, for the isotropic case, each polygon or polyhedron cannot be too “thin”.

In practice, one typically wants the mesh elements to be as regular in shape and uniform in size as possible under certain metrics, since the approximation error and the condition number of the discrete system depend closely on the mesh geometry. For simplicial meshes, this dependence has been well studied under the topic of “mesh quality measures” and many different computable quality measures have been designed; e.g., see [7, 21, 26, 31, 39]. More importantly, mesh quality measures often play an essential role in constructing efficient mesh adaptation algorithms since they also provide a definition of optimal meshes.

Traditionally, isotropic mesh adaptation has been studied, where the shape of mesh elements is kept as regular (in the Euclidean metric) as possible. On the other

*W.H. was supported in part by the NSF under Grant DMS-1115118 and the University of Kansas GRF Award # 2301056. Y.W. was supported by the FY2015 Spring Travel Program from the College of Art & Science, Oklahoma State University, during her visit to the University of Kansas in June 2015.

[†]Department of Mathematics, The University of Kansas, Lawrence, KS 66045. E-mail: whuang@ku.edu

[‡]Department of Mathematics, Oklahoma State University, Stillwater, OK 74078. E-mail: yanqiu.wang@okstate.edu

hand, anisotropic mesh adaptation allows elements to have large aspect ratio. The key is to keep elements to be aligned in some extent with the geometry of the solution. An anisotropic mesh can be viewed as a uniform one in a Riemannian metric, as in the \mathbb{M} -uniform mesh approach of anisotropic mesh generation [23]. It has been amply demonstrated that a properly generated anisotropic mesh can be much better aligned with the geometry of the solution and lead to a much more accurate solution than an isotropic mesh; e.g., see [2, 3, 6, 9, 11, 12, 21, 23, 37] among many other work. It is worth pointing out that all of these work is concerned with simplicial meshes. There do not seem to exist any systematic studies on anisotropic quality measures and adaptation for polygonal/polyhedral meshes.

The objective of this work is to develop anisotropic mesh quality measures for polygonal meshes and an anisotropic adaptive polygonal mesh method. An efficient and commonly used way to generate a polygonal (or polyhedral) mesh is through the Voronoi diagram. The latter is a partition of a spatial domain consisting of polygonal/polyhedral cells associated with a given set of points called “generators” or “seeds” such that each cell contains only points closer to its associated generator than to the others. Generally speaking, the quality of Voronoi meshes is not necessarily good. The most common strategy to obtain a high quality polygonal mesh is to start with a Voronoi mesh and then employ algorithms such as Lloyd’s algorithm to obtain a centroidal Voronoi tessellation (CVT) [10]. CVT can also be generated under a scalar metric function defining distance among points, i.e., a scalar weight function specifying the mesh density across the domain. This results in isotropic polygonal meshes, while our work here focuses on measuring and generating anisotropic polygonal meshes of high quality.

A major difference between triangular and polygonal meshes is that, all of the elements in a triangular mesh are affine similar to a single reference triangle and their quality in shape and size can be measured by comparing them to the reference element. Unfortunately, this does not work for polygonal meshes since elements with the same number of vertices cannot be mapped to a single element of the same number of vertices under affine mappings. The difference poses great difficulty in studying polygonal mesh quality measures, as one can no longer use many techniques developed for triangular meshes. In this work, in parallel to the so-called alignment (for regularity) and equidistribution (for size) measures for simplicial meshes [20, 21], we shall develop three sets of polygonal mesh alignment and equidistribution measures: one based on least squares fitting, one based on generalized barycentric mapping, and the other based on singular value decomposition of edge matrices. Numerical tests show that all three sets of mesh quality measures give good indicators of the actual anisotropic mesh quality under given metrics, with individual emphases on slightly different aspects. Based on one of these sets of quality measures and using the so-called Moving Mesh PDE (MMPDE) moving mesh method [26], we construct an anisotropic adaptive polygonal mesh method for the numerical solution of second order elliptic equations, with the aim of minimizing the L^2 norm of approximation errors. Numerical examples will be presented to demonstrate the effectiveness of the method.

The paper is organized as follows. Section 2 is devoted to the development of three sets of alignment and equidistribution quality measures for polygonal meshes. An anisotropic adaptive polygonal mesh method is constructed in Section 3 based on one set of the quality measures and the MMPDE moving mesh method. Numerical results obtained with the proposed method for two examples are presented in Section

4. Conclusions are drawn in the last section.

Throughout the paper, the short name “ n -gon” stands for a polygon with n vertices. We consider meshes consisting of convex polygons in this work. We also assume that all polygons in a mesh are non-degenerate, i.e., any three vertices of a polygon do not lie on the same line.

2. Anisotropic mesh quality measures. In this section we study three sets of mesh quality measures for a general polygonal mesh \mathcal{T} on a two-dimensional bounded, polygonal domain Ω under the metric specified by a given tensor \mathbb{M} . The metric tensor $\mathbb{M} = \mathbb{M}(\mathbf{x})$ is assumed to be symmetric and uniformly positive definite on Ω . It depends on the physical solution in the case of mesh adaptation for the numerical solution of PDEs (See Section 3). The simplest case is $\mathbb{M} = I$ (the Euclidean metric) where the mesh quality measures evaluate the shape regularity and size uniformity of the mesh.

2.1. Continuous-level mesh quality measures. Different from triangles, n -gons with $n > 3$ in a polygonal mesh generally are not affine similar to a single reference n -gon. The idea of measuring the quality of a mesh through comparing its elements to a single reference element does not work for polygonal meshes. Nevertheless, the idea can be used indirectly. In the following, we shall take a close look at the development of mesh quality measures for a triangular mesh and then establish continuous-level mesh quality measures which will serve as a unified framework for developing mesh quality measures for a polygonal mesh.

We assume that a computational mesh \mathcal{T}_C has been chosen. We emphasize that, in addition to a conventional mesh, \mathcal{T}_C can be chosen as a collection of polygons. The only requirement is that, for any $T \in \mathcal{T}$, there exists a corresponding reference polygon $T_C \in \mathcal{T}_C$ with an associated bijection $\mathcal{F}_T : T_C \rightarrow T$. The global bijection \mathcal{F} is the collection of all local \mathcal{F}_T 's, and we want it to be piecewise differentiable so that its Jacobian matrix exists almost everywhere. Generally speaking, \mathcal{T}_C should consist of good quality polygons under the Euclidean metric.

We now consider the mesh quality measures for a triangular mesh. To this end, we assume that both \mathcal{T} and \mathcal{T}_C are triangular meshes. Then, one can simply set $\mathcal{F}_T : T_C \rightarrow T$ to be an affine mapping and evaluate the quality of \mathcal{T} with reference to \mathcal{T}_C . Specifically, the triangles in \mathcal{T} are said to have a good shape if they, when measured in the metric \mathbb{M} , are similar to their counterparts in \mathcal{T}_C measured in the Euclidean metric. They are said to have the same size when the ratio of their area (measured in the metric \mathbb{M}) to the area of their counterparts in \mathcal{T}_C stays constant. Denote the vertices of T and T_C by \mathbf{x}_i and $\boldsymbol{\xi}_i$, $i = 1, 2, 3$, respectively. If we approximate the metric tensor \mathbb{M} by its average on T ,

$$\mathbb{M}_T = \frac{1}{|T|} \int_T \mathbb{M} d\mathbf{x}, \quad (2.1)$$

where $|T|$ is the area of T , then the similarity requirement between T and T_C can be expressed mathematically as

$$(\mathbf{x}_i - \mathbf{x}_j)^t \mathbb{M}_T (\mathbf{x}_i - \mathbf{x}_j) = \sigma_T (\boldsymbol{\xi}_i - \boldsymbol{\xi}_j)^t (\boldsymbol{\xi}_i - \boldsymbol{\xi}_j), \quad i, j = 1, 2, 3, i \neq j$$

where σ_T is a constant on T . Using the affine mapping $\mathcal{F}_T : T_C \rightarrow T$ and its Jacobian matrix \mathbb{J}_T (which is constant on T), we can rewrite the above condition as

$$(\boldsymbol{\xi}_i - \boldsymbol{\xi}_j)^t \mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T (\boldsymbol{\xi}_i - \boldsymbol{\xi}_j) = \sigma_T (\boldsymbol{\xi}_i - \boldsymbol{\xi}_j)^t (\boldsymbol{\xi}_i - \boldsymbol{\xi}_j), \quad i, j = 1, 2, 3, i \neq j.$$

It can be shown (cf. [26, Lemma 4.1.1]) that the above condition implies

$$\mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T = \sigma_T I. \quad (2.2)$$

This condition should apply to all triangles in \mathcal{T} for them to have a good shape with reference to their counterparts in \mathcal{T}_C . Taking determinant and square root of both sides of (2.2) and using the fact that $\det(\mathbb{J}_T) = |T|/|T_C|$, one gets

$$\frac{|T| \det(\mathbb{M}_T)^{\frac{1}{2}}}{|T_C|} = \sigma_T.$$

Then the uniform size requirement simply implies that σ_T should stay the same for all the triangles. By changing its subscript T to h , i.e., $\sigma_T \rightarrow \sigma_h$, we can rewrite the above equation into

$$|T| \det(\mathbb{M}_T)^{\frac{1}{2}} = \sigma_h |T_C|.$$

Summarizing it over all triangles, we can find σ_h as

$$\sigma_h = \frac{1}{|\mathcal{T}_C|} \sum_{T \in \mathcal{T}} |T| \sqrt{\det(\mathbb{M}_T)}, \quad (2.3)$$

where $|\mathcal{T}_C|$ denotes the total area of \mathcal{T}_C . Moreover, (2.2) becomes

$$\mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T = \sigma_h I, \quad \forall T \in \mathcal{T}. \quad (2.4)$$

From this the so-called alignment and equidistribution quality measures can be developed for a triangular mesh (cf. (2.10) below).

We now find a continuous analogue of (2.4). To this end, we define the computational domain as

$$\Omega_c = \bigcup_{T_C \in \mathcal{T}_C} T_C.$$

Note that Ω_c can be a conventional two-dimensional domain or a collection of polygons. It is not difficult to see that a continuous analogue of (2.4) is

$$\mathbb{J}^t \mathbb{M} \mathbb{J} = \sigma I, \quad \forall \mathbf{x} \in \Omega \quad (2.5)$$

where σ is a constant, $\mathbb{J} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ is the Jacobian matrix of the mapping $\mathcal{F} : \Omega_c \rightarrow \Omega$, and $\boldsymbol{\xi}$ and \mathbf{x} are the coordinates of Ω_c and Ω , respectively. At the continuous level, the mesh is represented by the mapping \mathcal{F} from Ω_c to Ω . Moreover, the constant σ in (2.5) is determined by compatibility. Indeed, taking determinant and square root on both sides of (2.5) and integrating the resultant equation over Ω_c , we have

$$\sigma = \frac{1}{|\Omega_c|} \int_{\Omega} \sqrt{\det(\mathbb{M})} d\mathbf{x}, \quad (2.6)$$

where $|\Omega_c|$ denotes the area of Ω_c .

The condition (2.5) implies that all of the eigenvalues of matrix $\mathbb{J}^t \mathbb{M} \mathbb{J}$ are equal to σ . The requirement is equivalent to the combination of two conditions: all of the eigenvalues are equal to each other and the square root of the product of the eigenvalues is equal to σ . These two conditions give rise to the alignment and equidistribution

conditions [21]. From the inequality of the arithmetic and geometric means it is not difficult to show that *the alignment condition at the continuous level*,

$$\frac{1}{2}\text{trace}(\mathbb{J}^t\mathbb{M}\mathbb{J}) = \det(\mathbb{J}^t\mathbb{M}\mathbb{J})^{1/2}, \quad (2.7)$$

is equivalent to requiring all of the eigenvalues of $\mathbb{J}^t\mathbb{M}\mathbb{J}$ to be equal to each other. A discrete version of this condition for a triangle T requires that T (measured in the metric \mathbb{M}_T) be similar to T_C .

On the other hand, notice that the product of the eigenvalues of $\mathbb{J}^t\mathbb{M}\mathbb{J}$ is equal to the determinant of the matrix. We can express the second condition (*the equidistribution condition at the continuous level*) mathematically as

$$\det(\mathbb{J})\sqrt{\det(\mathbb{M})} = \sigma. \quad (2.8)$$

For a triangular mesh, the left-hand side is proportional to the area of a triangle under the metric \mathbb{M} and the condition requires all of the triangles to have the same size.

Using (2.7) and (2.8), we can define the measuring functions as

$$q_{ali}(\mathbf{x}) = \frac{\text{trace}(\mathbb{J}^t\mathbb{M}\mathbb{J})}{2\det(\mathbb{J}^t\mathbb{M}\mathbb{J})^{1/2}}, \quad q_{eq}(\mathbf{x}) = \frac{\det(\mathbb{J})\sqrt{\det(\mathbb{M})}}{\sigma}. \quad (2.9)$$

In the ideal case when perfect alignment and equidistribution are reached everywhere, one has $q_{ali} \equiv 1$ and $q_{eq} \equiv 1$ over the entire Ω . For the general situation, we may not have perfect alignment or equidistribution. In this case, functions $q_{ali}(\mathbf{x})$ and $q_{eq}(\mathbf{x})$ indicate how closely the alignment and equidistribution conditions are satisfied at point \mathbf{x} . The overall alignment and distribution measures can then be defined as

$$Q_{ali} = \max_{\mathbf{x} \in \Omega} q_{ali}(\mathbf{x}), \quad Q_{eq} = \max_{\mathbf{x} \in \Omega} q_{eq}(\mathbf{x}). \quad (2.10)$$

From the inequality of the arithmetic and geometric means and the fact that

$$\int_{\Omega_c} q_{eq} d\boldsymbol{\xi} = |\Omega_c|,$$

it is not difficult to show that both Q_{ali} and Q_{eq} have the range $[1, \infty)$. Moreover, the smaller they are, the better the alignment and equidistribution of the mapping \mathcal{F} .

The above defined continuous level mesh quality measures will serve as a unified framework for defining actual mesh quality measures for a polygonal mesh. Clearly, Q_{ali} and Q_{eq} depend on the metric tensor \mathbb{M} and the computational mesh \mathcal{T}_C (which determines Ω_c). Recall that \mathcal{T}_C can be a conventional mesh or a collection of polygons and the corresponding Ω_c is a conventional domain or a collection of polygons. From (2.9), one can see that the determining factor for Q_{ali} and Q_{eq} is the Jacobian matrix \mathbb{J} of the bijection \mathcal{F} and how Ω_c looks has little weight on the definition. As long as there is a well-defined \mathbb{J} almost everywhere in Ω , Ω_c does not have to be a continuous manifold.

It is useful to point out that, the mapping \mathcal{F} is defined through the local mapping $\mathcal{F}_T : T_C \rightarrow T$ for all $T \in \mathcal{T}$. Unlike triangular meshes, there does not exist an affine mapping between T and T_C in general for polygonal meshes. Thus, there are many ways to specify \mathcal{F} for a given polygonal mesh \mathcal{T} . Commonly, \mathcal{F}_T is chosen to be a

non-affine mapping for a general T_C or an affine mapping for a specially chosen T_C ; see the next three subsections.

We now study three discretizations/discrete approximations to the continuous mesh quality measures. The key components are the choices of \mathcal{T}_C and the corresponding mapping \mathcal{F} (to represent the mesh \mathcal{T}).

2.2. Approximation 1: use least squares fitting and edge matrices. In this approximation, we assume that the computational mesh \mathcal{T}_C has been chosen to be a conventional mesh or a collection of reference polygons. Typically \mathcal{T}_C should consist of good quality polygons in the Euclidean metric. These good quality polygons can be regular n -gons or almost regular n -gons.

As part of the definition, we choose to approximate the metric tensor \mathbb{M} on T by its average (2.1). This approximation is reasonably accurate while making \mathbb{M} be constant on each element. To have a constant \mathbb{M} on each element is important since it can significantly simplify the analysis and derivation.

As mentioned before, \mathcal{F} is defined through the local mapping $\mathcal{F}_T : T_C \rightarrow T$. The existence of such a local mapping is guaranteed by the Riemann mapping theorem, which states that there exists a conformal mapping between any two n -gons. As we will see in the next subsection, there actually exist many such mappings. They all satisfy

$$\mathbf{x}_i = \mathcal{F}_T(\boldsymbol{\xi}_i), \quad i = 1, \dots, n \quad (2.11)$$

where $\boldsymbol{\xi}_i$ and \mathbf{x}_i , $i = 1, \dots, n$, are the vertices of T_C and T , respectively. Instead of constructing a specific example of these mappings, we consider to fit an affine mapping $A_T \boldsymbol{\xi} + \mathbf{c}$ in the least squares sense based on the above conditions. Note that such a mapping is an approximation to all bijections between \mathcal{T}_C and Ω satisfying (2.11). Although the approximation is very rough, it serves our purpose to define mesh quality measures. The condition for the least squares fitting is

$$A_T \boldsymbol{\xi}_i + \mathbf{c} = \mathbf{x}_i, \quad i = 1, \dots, n.$$

Subtracting the first equation from the others, we get

$$A_T(\boldsymbol{\xi}_i - \boldsymbol{\xi}_1) = \mathbf{x}_i - \mathbf{x}_1, \quad i = 2, \dots, n.$$

A least squares solution for the above condition is

$$A_T = E_T E_{T_C}^+,$$

where E_T and E_{T_C} are the edge matrices defined by

$$\begin{aligned} E_T &= [\mathbf{x}_2 - \mathbf{x}_1, \dots, \mathbf{x}_n - \mathbf{x}_1] \in \mathbb{R}^{2 \times (n-1)}, \\ E_{T_C} &= [\boldsymbol{\xi}_2 - \boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n - \boldsymbol{\xi}_1] \in \mathbb{R}^{2 \times (n-1)}, \end{aligned}$$

and $E_{T_C}^+$ is the pseudo-inverse of E_{T_C} . Since both T and T_C are convex and non-degenerate, both E_T and E_{T_C} have full row rank. Then, the pseudo-inverse of E_{T_C} can be found as

$$E_{T_C}^+ = E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1} \in \mathbb{R}^{(n-1) \times 2},$$

which gives rise to

$$A_T = E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}. \quad (2.12)$$

By construction, A_T is an approximation to the Jacobian matrix of any piecewise continuously differentiable bijection between \mathcal{T}_C and \mathcal{T} satisfying (2.11).

Replacing \mathbb{J} and \mathbb{M} by A_T and \mathbb{M}_T in (2.9) and (2.10), we obtain the first set of mesh quality measures for a polygonal mesh,

$$Q_{ali,1} = \max_{T \in \mathcal{T}} \frac{\text{trace}([E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}]^t \mathbb{M}_T [E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}])}{2 \det([E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}]^t \mathbb{M}_T [E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}])^{1/2}}, \quad (2.13)$$

$$Q_{eq,1} = \max_{T \in \mathcal{T}} \frac{\det(E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}) \sqrt{\det(\mathbb{M}_T)}}{\sigma_{h,1}}, \quad (2.14)$$

where

$$\sigma_{h,1} = \frac{1}{N_p} \sum_{T \in \mathcal{T}} \det(E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}) \sqrt{\det(\mathbb{M}_T)} \quad (2.15)$$

and N_p is the number of the polygons in \mathcal{T} . Notice that $\sigma_{h,1}$ is not a direct approximation of σ defined in (2.6). Instead, it is defined by summarizing the following discrete equidistribution condition over all polygons,

$$\det(E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}) \sqrt{\det(\mathbb{M}_T)} = \sigma_{h,1}.$$

The same strategy will be used in defining other two sets of mesh quality measures without explicit mentioning. This also makes the current formulas consistent with those developed for simplicial meshes in [21, 26].

To show how well $Q_{ali,1}$ and $Q_{eq,1}$ work, we present numerical results obtained with Lloyd's algorithm for centroidal Voronoi tessellations (CVTs) [32]. The algorithm is known to produce a sequence of Voronoi meshes that converges to a CVT. We expect to see that both $Q_{ali,1}$ and $Q_{eq,1}$ (with $\mathbb{M} = I$) for the sequence of meshes decrease and converge to one if they are correct indicators for the shape regularity and size uniformity of the Voronoi meshes.

In Fig. 2.1, we show Voronoi meshes of 8×8 , 16×16 and 32×32 cells obtained with Lloyd's algorithm. Visually we can see that the meshes are becoming better in the sense that the cells are getting more regular in shape and more uniform in size. To compute $Q_{ali,1}$ and $Q_{eq,1}$, we choose the unitary regular n -gon with vertices $\xi_i = (\cos \frac{2\pi i}{n}, \sin \frac{2\pi i}{n})^t$, $i = 1, \dots, n$ as T_C for each n -gon in T . The results are reported in Fig. 2.2. One can see that both $Q_{ali,1}$ and $Q_{eq,1}$ decrease towards one as the number of iterations increases. This reflects the convergence nature of Lloyd's algorithm. Moreover, the decrease is more significant in the first few iterations and then getting much slower. This confirms the fact that Lloyd's algorithm is efficient for obtaining a CVT of reasonably good quality but very slow for a CVT with high accuracy. Furthermore, one may notice that the decrease of $Q_{ali,1}$ and $Q_{eq,1}$ is not monotone. This may be attributed to the facts that (a) Lloyd's algorithm is not designed specifically to minimize these quality measures and (b) $Q_{ali,1}$ and $Q_{eq,1}$ measure the quality of worst mesh elements. Overall, we see that $Q_{ali,1}$ and $Q_{eq,1}$ correctly reflect the polygonal mesh quality under the identity metric.

2.3. Approximation 2: use generalized barycentric mappings. This approximation is similar to Approximation 1 except that here we construct a specific local mapping $\mathcal{F}_T : T_C \rightarrow T$ using generalized barycentric mappings [1, 16, 29, 38]. The generalized barycentric mappings are related to generalized barycentric coordinates (GBCs) (see [13, 14, 15, 46] and references therein) which are defined below.

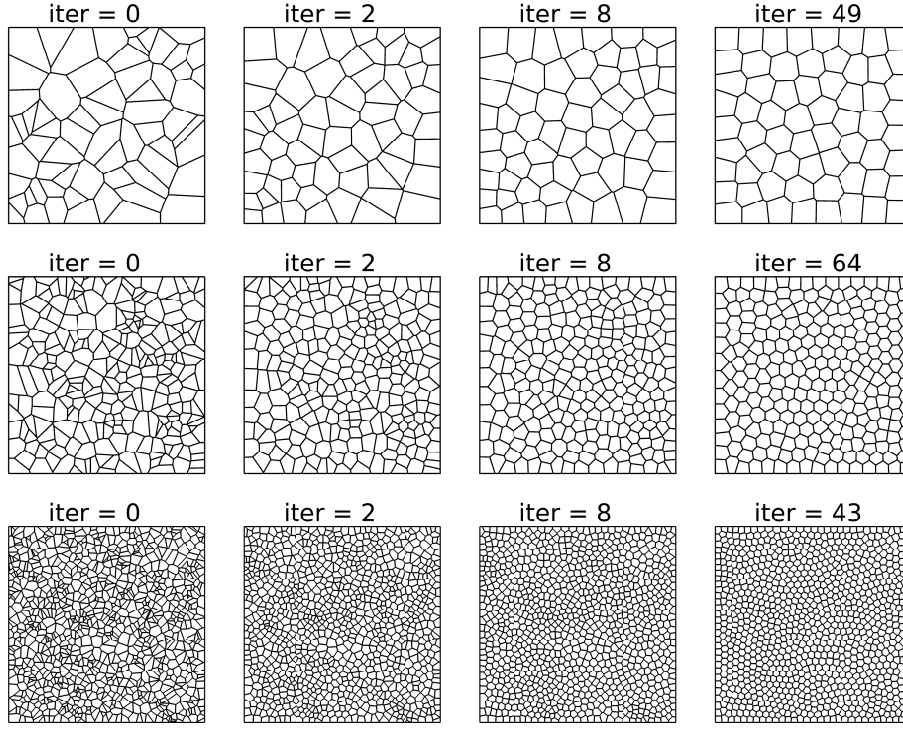


Fig. 2.1: Voronoi meshes of 8×8 , 16×16 , and 32×32 cells in Lloyd's iteration.

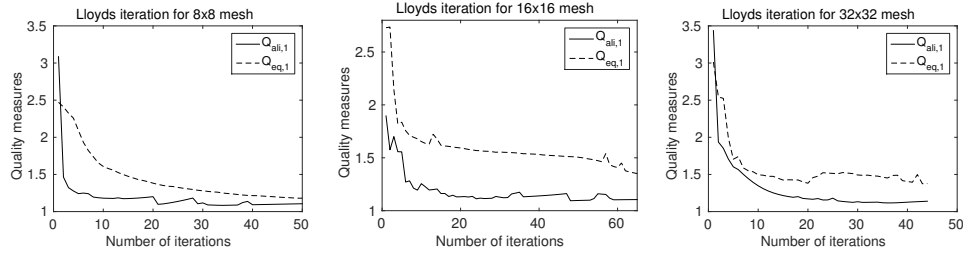


Fig. 2.2: History of $Q_{ali,1}$ and $Q_{eq,1}$ in Lloyd's iteration for meshes with 8×8 , 16×16 and 32×32 cells.

DEFINITION 2.1. *The generalized barycentric coordinates for a given n -gon T are the functions $\lambda_i : T \rightarrow \mathbb{R}$, $i = 1, \dots, n$ satisfying*

- (i) *(Non-negativity) All $\lambda_i(\mathbf{x})$ for $1 \leq i \leq n$ are non-negative on T ;*
- (ii) *(Linear precision) There hold*

$$\sum_{i=1}^n \lambda_i(\mathbf{x}) = 1, \quad \sum_{i=1}^n \lambda_i(\mathbf{x}) \mathbf{x}_i = \mathbf{x}, \quad \forall \mathbf{x} \in T.$$

Take a set of GBCs on T_C and denote it by $(\lambda_1(\boldsymbol{\xi}), \lambda_2(\boldsymbol{\xi}), \dots, \lambda_n(\boldsymbol{\xi}))$. We can

define a mapping \mathcal{F}_T from T_C to T as

$$\mathbf{x} = \mathcal{F}_T(\boldsymbol{\xi}) \triangleq \sum_{i=1}^n \lambda_i(\boldsymbol{\xi}) \mathbf{x}_i, \quad \forall \boldsymbol{\xi} \in T_C.$$

Such a mapping is called a generalized barycentric mapping in literature and has important applications in several fields. An important non-trivial question is whether \mathcal{F}_T defines a bijection or not. Fortunately, it has been answered positively for two types of GBCs on convex polygons, Wachspress coordinates [16] and piecewise linear coordinates [1, 29].

The Wachspress coordinates [34, 46] are defined as

$$\lambda_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j=1}^n w_j(\mathbf{x})}, \quad i = 1, \dots, n$$

where

$$w_i(\mathbf{x}) = \overline{\mathbf{x}_{i-1} \mathbf{x}_i \mathbf{x}_{i+1}} \prod_{j \neq i, i-1} \overline{\mathbf{x} \mathbf{x}_j \mathbf{x}_{j+1}}, \quad i = 1, \dots, n$$

and $\overline{\mathbf{x} \mathbf{y} \mathbf{z}}$ denotes the signed area of the triangle with vertices \mathbf{x} , \mathbf{y} and \mathbf{z} . The Wachspress coordinates are linear for $n = 3$, bilinear for $n = 4$, and rational for general n .

We now consider piecewise linear coordinates. Let \mathcal{T}_{T_C} be a triangulation of T_C . Let λ_i , $i = 1, \dots, n$, be piecewise linear functions on \mathcal{T}_{T_C} satisfying $\lambda_i(\boldsymbol{\xi}_j) = \delta_{ij}$ and $\sum_{i=1}^n \lambda_i = 1$. This defines the piecewise linear GBCs associated with \mathcal{T}_{T_C} . Note that a different triangulation of T_C can lead to a different set of GBCs. Moreover, \mathcal{T}_{T_C} can be a triangulation using exactly the same vertices of T_C , or it can have extra vertices added as long as the extra vertices have their own and distinct barycentric coordinates specified. Furthermore, the piecewise linear bijection \mathcal{F}_T from T_C to T can be obtained in a different but equivalent way: specify the same type of triangulations for T_C and T and then use piecewise linear mappings to map each individual triangle in T_C to the corresponding triangle in T .

In practice, the piecewise linear barycentric mapping is much easier to compute than the Wachspress barycentric mapping. For this reason, we consider the former only in this work. In this case, the Jacobian matrix \mathbb{J}_T of the mapping \mathcal{F}_T is piecewise constant on each $T \in \mathcal{T}$. If we also use a piecewise constant approximation of \mathbb{M} on T , from (2.9) and (2.10) we obtain the second set of mesh quality measures as

$$Q_{ali,2} = \max_{T \in \mathcal{T}} \max_{K \in \mathcal{T}_T} \frac{\text{trace}((\mathbb{J}_T|_K)^t \mathbb{M}_K \mathbb{J}_T|_K)}{2 \det((\mathbb{J}_T|_K)^t \mathbb{M}_K \mathbb{J}_T|_K)^{1/2}}, \quad (2.16)$$

$$Q_{eq,2} = \max_{T \in \mathcal{T}} \max_{K \in \mathcal{T}_T} \frac{\det((\mathbb{J}_T|_K)^t \mathbb{M}_K \mathbb{J}_T|_K) \sqrt{\det(\mathbb{M}_K)}}{\sigma_{h,2}}, \quad (2.17)$$

where $\mathbb{J}_T|_K$ is the restriction of \mathbb{J}_T on K ,

$$\sigma_{h,2} = \frac{1}{N_{tri}} \sum_{T \in \mathcal{T}} \sum_{K \in \mathcal{T}_T} \det((\mathbb{J}_T|_K)^t \mathbb{M}_K \mathbb{J}_T|_K) \sqrt{\det(\mathbb{M}_K)} \quad (2.18)$$

and N_{tri} is the total number of the triangles in the mesh.

As for Approximation 1, we test the second set of the mesh quality measures on the same Lloyd iterations shown in Fig. 2.1. Again, each n -gon is compared with a regular reference n -gon. There are two easy ways to define the triangular subdivision on T (as well as on T_C): (a) connect \mathbf{x}_1 with all other vertices of T and get a triangulation; (b) connect all vertices to the arithmetic mean of vertices. Since T is convex, the arithmetic mean of its vertices lies inside T . We test both subdivisions. The results are reported in Figs. 2.3 and 2.4.

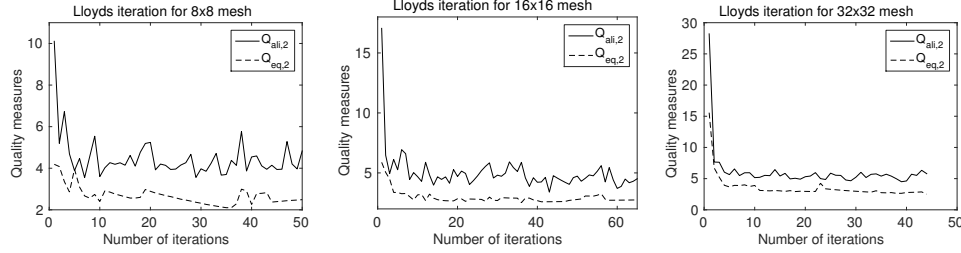


Fig. 2.3: History of $Q_{ali,2}$ and $Q_{eq,2}$ in Lloyd's iteration for meshes with 8×8 , 16×16 and 32×32 cells. Here, the triangular subdivision (a) is used.

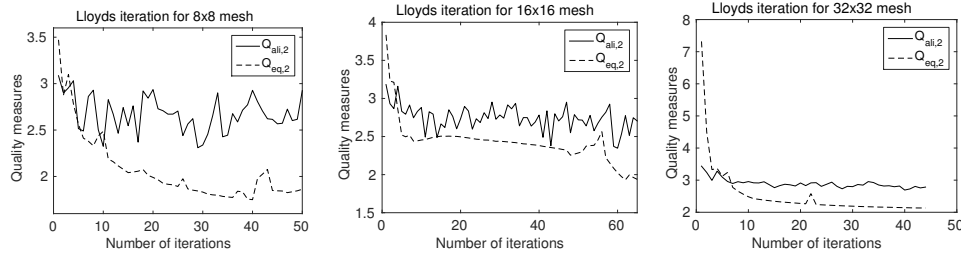


Fig. 2.4: History of $Q_{ali,2}$ and $Q_{eq,2}$ in Lloyd's iteration for meshes with 8×8 , 16×16 and 32×32 cells. Here, the triangular subdivision (b) is used.

$Q_{ali,2}$ and $Q_{eq,2}$ have similar overall decreases, more dramatic drops at the first few iterations, and not monotone decreases as the mesh quality measures in Approximation 1. They also provide correct indications of the behavior of Lloyd's algorithm.

On the other hand, one may notice that the values of $Q_{ali,2}$ and $Q_{eq,2}$ are generally greater than $Q_{ali,1}$ and $Q_{eq,1}$ and more sensitive to quality changes (with more and stronger oscillations). This means that those Voronoi meshes have worse quality in the former's eyes than in the latter's eyes. The reason why $Q_{ali,2}$ and $Q_{eq,2}$ are pickier is due to their construction: they favor triangulations that look good for the used piecewise generalized barycentric mapping. To explain this, we examine a random n -gon T in a CVT, as shown in Fig. 2.5. Although T is considered as of good shape by Lloyd's algorithm, i.e., its generator is close to its barycenter, T may still contain short edges. When both the regular reference n -gon and T are cut into triangles which are compared one-by-one, the triangles associated with the short edges have a bad shape.

It is also interesting to point out that subdivision (b) gives smaller $Q_{ali,2}$ and $Q_{eq,2}$ than subdivision (a).

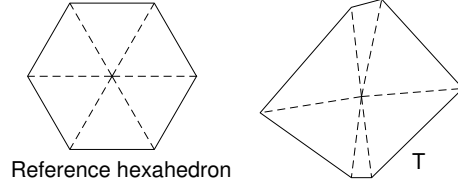


Fig. 2.5: Piecewise linear barycentric mapping from a regular reference hexahedron to T using triangular subdivision (b). Because of the short edges in T , some sub-triangles of T can have bad shape, which affects the value of $Q_{ali,2}$.

2.4. Approximation 3: employ a special design that uses infinitely many reference n -gons to ensure \mathcal{F}_T be affine. This approximation is different from the previous two. Here, we introduce a set of infinitely many reference polygons of relatively good shape. Such a large set of reference polygons allows us to use affine mappings to define \mathcal{F}_T . An important tool for the development is the connection between an arbitrary n -gon and a high dimensional simplex. Such a connection has been known in geometry and discrete mathematics (e.g., see [49, Chapter 0]). Here we use it to analyze polygonal mesh quality.

Consider a set of GBCs λ_i , $i = 1, \dots, n$, on $T \in \mathcal{T}$. Recall that

$$\mathbf{x} = \sum_{i=1}^n \lambda_i \mathbf{x}_i, \quad \forall \mathbf{x} \in T \quad (2.19)$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]^t$ lies in the set

$$S_{n-1} = \{[\lambda_1, \lambda_2, \dots, \lambda_n]^t \in \mathbb{R}^n \mid \sum_{i=1}^n \lambda_i = 1 \text{ and } \lambda_i \geq 0, 1 \leq i \leq n\}.$$

The set S_{n-1} is a regular $(n-1)$ -simplex in \mathbb{R}^n , meaning that it has complete symmetry with regard to transitions over vertices, edges, and higher dimensional faces, etc. S_{n-1} is contained in an $(n-1)$ -dimensional hyperplane $\sum_{i=1}^n \lambda_i = 1$ in \mathbb{R}^n and is essentially an $(n-1)$ -dimensional simplex living in the n -dimensional space.

Equation (2.19) can be viewed as a linear mapping from S_{n-1} onto T . We shall add a translation to this mapping to make the center of S_{n-1} mapped to the origin, which will greatly simplify the derivation. Denote the arithmetic mean of all vertices of T , which will be called as the *arithmetic center* of T , by

$$\mathbf{x}_T = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i.$$

The arithmetic center of a convex polygon always locates inside the polygon. Equation (2.19) can be rewritten into

$$\mathbf{x} - \mathbf{x}_T = \sum_{i=1}^n \lambda_i (\mathbf{x}_i - \mathbf{x}_T). \quad (2.20)$$

Denote by $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x}_T$ a translation in coordinates which moves \mathbf{x}_T to the origin. Then, (2.20) can be viewed as a linear mapping from $\boldsymbol{\lambda}$ to $\hat{\mathbf{x}}$, with the geometric

center of S_{n-1} , denoted by $\boldsymbol{\lambda}_c = [\frac{1}{n}, \dots, \frac{1}{n}]^t$, being mapped into the origin $\hat{\mathbf{x}} = \mathbf{0}$. For simplicity and with abusing of the notation, it is convenient to still denote the coordinate for T by \mathbf{x} , which should actually be understood as $\hat{\mathbf{x}}$, together with the assumption that $\mathbf{x}_T = \mathbf{0}$.

The mapping from S_{n-1} to T can be written as

$$S_{n-1} \xrightarrow{\Psi_T} T, \quad \Psi_T(\boldsymbol{\lambda}) = B_T \boldsymbol{\lambda} \triangleq [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_n] \boldsymbol{\lambda}. \quad (2.21)$$

Clearly, one has

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{x}_T = \Psi_T(\boldsymbol{\lambda}_c).$$

When $n = 3$, (2.21) together with the constraint $\sum_{i=1}^n \lambda_i = 1$ gives a uniquely solvable 3×3 linear system for non-degenerate T ,

$$\begin{bmatrix} B_T \\ \mathbf{1}^t \end{bmatrix} \boldsymbol{\lambda} \triangleq \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \\ 1 & 1 & 1 \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix},$$

i.e., the mapping Ψ_T is invertible. But for $n > 3$, the mapping Ψ_T is not invertible as there is a non-trivial kernel. This can also be explained by the fact that T is a 2-manifold while S_{n-1} is an $(n-1)$ -manifold.

To understand the linear mapping Ψ_T from S_{n-1} to T , we consider the singular value decomposition of the matrix $B_T \in \mathbb{R}^{2 \times n}$,

$$B_T = U_T \Sigma_T V_T^t,$$

where

$$\begin{aligned} U_T &= [\mathbf{u}_{1,T} \quad \mathbf{u}_{2,T}] \in \mathbb{R}^{2 \times 2}, & \Sigma_T &= \begin{bmatrix} \sigma_{1,T} & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{2,T} & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{2 \times n}, \\ V_T &= [\mathbf{v}_{1,T} \quad \mathbf{v}_{2,T} \quad \cdots \quad \mathbf{v}_{n,T}] \in \mathbb{R}^{n \times n}, \end{aligned}$$

$\sigma_{1,T} \geq \sigma_{2,T}$ are the singular values, and U_T and V_T are orthogonal matrices. The singular values are non-zero as long as the polygon T does not degenerate into a line segment. We further decompose

$$\Sigma_T = \begin{bmatrix} \sigma_{1,T} & 0 \\ 0 & \sigma_{2,T} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \triangleq D_T Q_T.$$

Then, the singular value decomposition of B_T can be rewritten into

$$B_T = (U_T D_T U_T^t)(U_T Q_T V_T^t) \triangleq \mathbb{J}_T P_T.$$

Now, the geometric meaning of the linear mapping Ψ_T (with coefficient matrix B_T) can be explained as follows (see an illustration given in Fig. 2.6).

First, the linear mapping Ψ_T applies to S_{n-1} with a linear transformation specified by

$$P_T = U_T Q_T V_T^t = U_T \begin{bmatrix} \mathbf{v}_{1,T}^t \\ \mathbf{v}_{2,T}^t \end{bmatrix}.$$

Since $Q_T V_T^t$ is an orthogonal projection from S_{n-1} onto the plane spanned by $\mathbf{v}_{1,T}$ and $\mathbf{v}_{2,T}$, P_T is an orthogonal projection followed by a 2D rotation/reflection U_T . The result is an n -gon

$$K_T = U_T \begin{bmatrix} \mathbf{v}_{1,T}^t \\ \mathbf{v}_{2,T}^t \end{bmatrix} (S_{n-1}). \quad (2.22)$$

Second, the mapping applies to K_T with a linear transformation $\mathcal{F}_T(\boldsymbol{\xi}) = \mathbb{J}_T \boldsymbol{\xi} = U_T D_T U_T^t \boldsymbol{\xi}$, resulting in $T = \mathcal{F}_T(K_T)$. Note that \mathbb{J}_T is symmetric and positive definite. Thus \mathcal{F}_T is an *anisotropic scaling*, i.e., scaling by factors $\sigma_{1,T}$ and $\sigma_{2,T}$ in the directions of $\mathbf{u}_{1,T}$ and $\mathbf{u}_{2,T}$, respectively. Polygons T and K_T are affine similar under the anisotropic scaling \mathcal{F}_T .

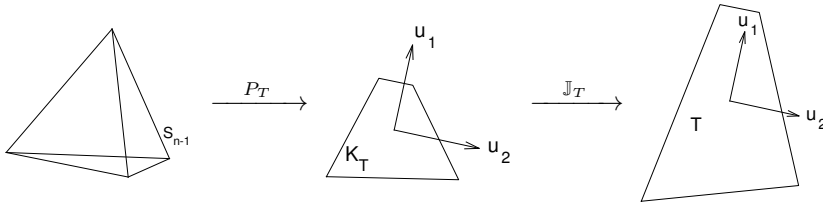


Fig. 2.6: Illustration of the linear mapping from simplex S_{n-1} to a polygon, which is decomposed into two affine mappings.

In the following we show that K_T has several nice properties. First of all, because T and K_T are affine similar to each other, we have the following three propositions.

PROPOSITION 2.2. *T is convex if and only if K_T is convex.*

PROPOSITION 2.3. *T is non-degenerate if and only if K_T is non-degenerate.*

PROPOSITION 2.4. *The arithmetic center of K_T locates at the origin.*

More importantly, K_T has a relatively good shape and can serve as a reference n -gon. To show this, we first need to clarify what it means for an n -gon to have a relatively good shape. For triangles, it is common to use the ratio between the radii of its inscribed circle and circumscribed circle as a measure. But an n -gon with $n \geq 4$ may not have inscribed or circumscribed circles. Therefore, we introduce the following definition.

DEFINITION 2.5. *For a given polygon T , its in-radius is the maximum radius of all disks contained inside \bar{T} , and its outer-radius is the minimum radius of all disks containing T .*

PROPOSITION 2.6. *Let T be a convex n -gon and K_T be defined in (2.22). Then, the in-radius of K_T is greater than or equal to $\sqrt{\frac{1}{n(n-1)}}$, and the outer-radius of K_T is less than or equal to $\sqrt{\frac{n-1}{n}}$. Consequently, the ratio ρ between the in-radius and the outer-radius of K_T is bounded by*

$$\frac{1}{n-1} \leq \rho \leq 1,$$

which depends on n but not on the shape of K_T .

Proof. Recall that S_{n-1} lies inside the hyperplane $\sum_{i=1}^n \lambda_i = 1$. By Proposition 2.4 and linearity, one has

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = P_T \boldsymbol{\lambda}_c = U_T \begin{bmatrix} \mathbf{v}_{1,T} \cdot \boldsymbol{\lambda}_c \\ \mathbf{v}_{2,T} \cdot \boldsymbol{\lambda}_c \end{bmatrix},$$

which implies that $\mathbf{v}_{1,T} \cdot \boldsymbol{\lambda}_c = \mathbf{v}_{2,T} \cdot \boldsymbol{\lambda}_c = 0$. Consequently, vectors in $\text{span}\{\mathbf{v}_{1,T}, \mathbf{v}_{2,T}\}$ are orthogonal to $\boldsymbol{\lambda}_c = [\frac{1}{n}, \dots, \frac{1}{n}]^t$, which is also the normal direction of the hyperplane $\sum_{i=1}^n \lambda_i = 0$. In other words, the 2D plane spanned by $\mathbf{v}_{1,T}$ and $\mathbf{v}_{2,T}$ lies inside the hyperplane $\sum_{i=1}^n \lambda_i = 0$, which is parallel to the hyperplane $\sum_{i=1}^n \lambda_i = 1$. This is important as it guarantees that the orthogonal projection of any ball inside the hyperplane $\sum_{i=1}^n \lambda_i = 1$ into $\text{span}\{\mathbf{v}_{1,T}, \mathbf{v}_{2,T}\}$ must be a circular disk, as illustrated in Fig. 2.7.

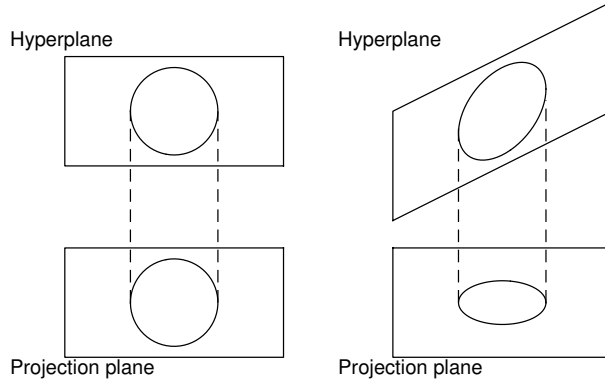


Fig. 2.7: Orthogonal projection from a hyperplane to a 2D plane. Left: If the 2D plane lies inside a parallel hyperplane, then projection of any ball in the hyperplane gives a circular disk; Right: if not parallel, the projection becomes an ellipse.

By Proposition 2.2, polygon K_T is also convex. Hence the projection P_T maps the inscribed ball of S_{n-1} into an inner disk of K_T , and the circumscribed ball of S_{n-1} into an outer disk of K_T . One simply needs to compute the radii of the inscribed and the circumscribed ball of S_{n-1} , which are just $\sqrt{\frac{1}{n(n-1)}}$ and $\sqrt{\frac{n-1}{n}}$. This completes the proof of the lemma. \square

REMARK 2.7. For $n > 3$, the projection of the inscribed (circumscribed) ball of S_{n-1} under P_T is not necessarily the largest disk in K_T (the smallest disk containing K_T).

Because of the uniform bound for ρ stated in Proposition 2.6, K_T has a relatively good shape and can thus be used as a reference n -gon.

A few possible K_T 's are shown in Fig. 2.8. When $n = 3$, the simplex S_2 is an equilateral triangle lying in the plane $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and the condition $\mathbf{v}_1 \cdot \mathbf{1} = \mathbf{v}_2 \cdot \mathbf{1} = 0$ implies that $\text{span}\{\mathbf{v}_1, \mathbf{v}_2\}$ is just the plane $\lambda_1 + \lambda_2 + \lambda_3 = 0$. Thus the reference 3-gons, or the reference triangles, are simply rotated/reflected equilateral triangles with edge length $\sqrt{2}$, as shown in Fig. 2.8. In other words, the (equilateral) reference triangle is unique up to rotation/reflection. When $n > 3$, on the other hand, the reference n -gons will no longer be unique even after rotation/reflection. However,

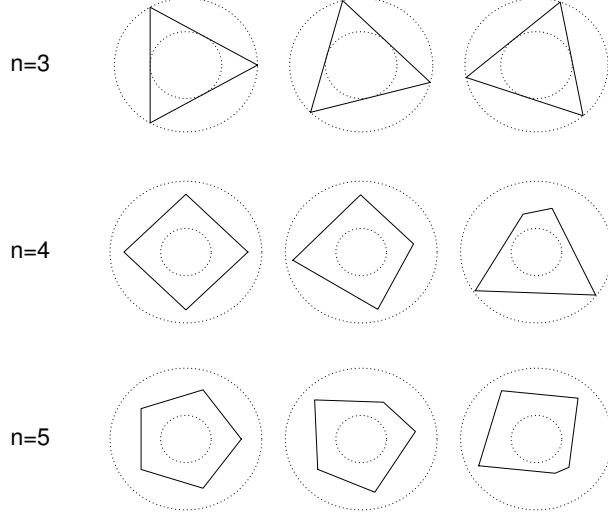


Fig. 2.8: Examples of reference polygons. The dotted circles are the projection of the inscribed and circumscribed balls of S_{n-1} under P_T .

they should all lie between two circles with radii $\sqrt{\frac{1}{n(n-1)}}$ and $\sqrt{\frac{n-1}{n}}$, as stated in Proposition 2.6; and they should all be convex and non-degenerate, as verified in propositions 2.2 and 2.3.

Now we are ready to define the mesh quality measures using the reference n -gons. Recall that each T has a unique K_T and \mathcal{T}_C is the collection of all those K_T 's. (We do not need to compute K_T 's when evaluating the mesh quality measures.) The Jacobian matrix \mathbb{J}_T of \mathcal{F}_T is defined as

$$\mathbb{J}_T = U_T \begin{bmatrix} \sigma_{1,T} & 0 \\ 0 & \sigma_{2,T} \end{bmatrix} U_T^t, \quad (2.23)$$

where U_T and $\sigma_{1,T}$ and $\sigma_{2,T}$ are the left singular vectors and singular values of the matrix $B_T = [\mathbf{x}_1 - \mathbf{x}_T, \dots, \mathbf{x}_n - \mathbf{x}_T]$. (Here we put back the translation just for convenience.) Using the metric tensor approximation (2.1), we can then define the mesh quality measures as

$$Q_{ali,3} = \max_{T \in \mathcal{T}} \frac{\text{trace}(\mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T)}{2 \det(\mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T)^{1/2}}, \quad (2.24)$$

$$Q_{eq,3} = \max_{T \in \mathcal{T}} \frac{\det(\mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T) \sqrt{\det(\mathbb{M}_T)}}{\sigma_{h,3}}, \quad (2.25)$$

where \mathbb{J}_T is defined in (2.23),

$$\sigma_{h,3} = \frac{1}{N_p} \sum_{T \in \mathcal{T}} \det(\mathbb{J}_T^t \mathbb{M}_T \mathbb{J}_T) \sqrt{\det(\mathbb{M}_T)}, \quad (2.26)$$

and N_p is the number of the polygons in \mathcal{T} .

We have tested the third set of mesh quality measures numerically on the same Lloyd iterations shown in Fig. 2.1. The results are reported in Fig. 2.9. Comparing

with the numerical results for $Q_{ali,1}$ and $Q_{eq,1}$ (cf. Fig. 2.2 and Table 2.1), one may notice that $Q_{ali,3}$ is slightly smaller than $Q_{ali,1}$ but $Q_{eq,3}$ is slightly larger than $Q_{eq,1}$ for the same mesh. This is because in the computation for the first approximation, regular n -gons are used as reference n -gons, which gives a precise control of the size of reference n -gons; while in the third approximation, the size of reference n -gons varies a bit as can be seen in Fig. 2.8. Overall, $Q_{ali,3}$ and $Q_{eq,3}$ also provide a good measure on the quality of Voronoi meshes. A major advantage of this approximation is that all \mathcal{F}_T 's are affine, which will be useful in error estimation in numerical solution of partial differential equations.

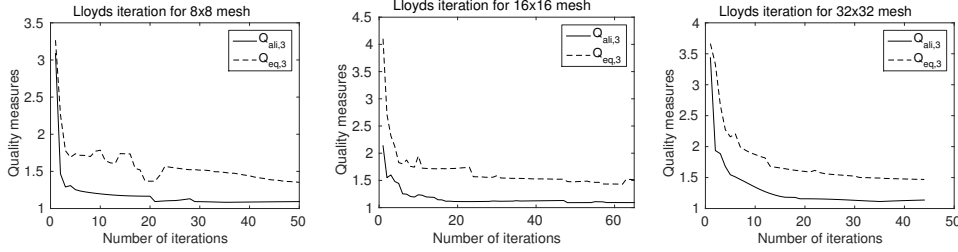


Fig. 2.9: History of $Q_{ali,3}$ and $Q_{eq,3}$ in Lloyd's iteration for meshes with 8×8 , 16×16 and 32×32 cells.

2.5. Summary. The three sets of mesh quality measures discussed in the previous subsections can more or less be viewed as discretizations or numerical approximations of the continuous level alignment and equidistribution quality measures defined in (2.10). Moreover, they all adopt the idea of evaluating the quality of a polygonal mesh by comparing its elements to their counterparts in the computational mesh which can be a conventional mesh or a collection of reference polygons. Numerical experiment shows that they all provide correct measures for the quality of Voronoi meshes generated by Lloyd's algorithm. Some of the main features of these mesh quality measures are elaborated in the following.

- $Q_{ali,3}$ (2.24) and $Q_{eq,3}$ (2.25) are fully determined by the mesh \mathcal{T} . Specifically, the computational mesh \mathcal{T}_C is a collection of K_T 's which are shown to have reasonably good quality and determined by the coordinates of the vertices of T . The mapping \mathcal{F}_T is an affine mapping from K_T to T , which is also determined by the coordinates of the vertices of T .
- $Q_{ali,1}$ (2.13) and $Q_{eq,1}$ (2.14) are not fully determined by the mesh \mathcal{T} . The computational mesh \mathcal{T}_C can be chosen by the user to be a mesh or a collection of reference n -gons. The measurement of the quality of \mathcal{T} depends on the choice of \mathcal{T}_C . The mapping \mathcal{F}_T is approximated by an affine mapping that is obtained by least squares fitting to the correspondence between the vertices of $T \in \mathcal{T}$ and $T_C \in \mathcal{T}_C$ and thus fully determined by \mathcal{T} and \mathcal{T}_C .
- $Q_{ali,2}$ (2.16) and $Q_{eq,2}$ (2.17) are not fully determined by the mesh \mathcal{T} . The computational mesh \mathcal{T}_C can be chosen by the user to be a mesh or a collection of reference n -gons. The mapping \mathcal{F}_T is specified using generalized barycentric mappings such as Wachspress and piecewise barycentric coordinates. The measurement of the quality of \mathcal{T} depends on the choice of \mathcal{T}_C as well as the choice of \mathcal{F}_T . The tests using Voronoi meshes generated by Lloyd's algorithm (cf. Table 2.1) suggest that $Q_{ali,2}$ and $Q_{eq,2}$ provide the toughest measures

(which give biggest values and are hardest to satisfy) among the three sets of mesh quality measures.

Table 2.1: Q_{ali} and Q_{eq} at selected Lloyd’s iteration steps for the 32×32 mesh. Subdivision (b) was used for $Q_{ali,2}$ and $Q_{eq,2}$.

Iter.	$Q_{ali,1}$	$Q_{eq,1}$	$Q_{ali,2}$	$Q_{eq,2}$	$Q_{ali,3}$	$Q_{eq,3}$
0	3.4362	3.0130	3.4362	7.3202	3.4362	3.6641
2	1.8519	2.5239	2.9940	3.3286	1.8851	2.6679
8	1.3927	1.5355	2.9190	2.5647	1.3869	1.9022
43	1.1394	1.3771	2.7847	2.1333	1.1370	1.4703

3. Anisotropic polygonal mesh adaptation. In this section, we study the anisotropic adaptation for polygonal meshes through a moving mesh method based on the so-called MMPDE (moving mesh PDE) [25, 26]. Notice that a mesh is completely determined by two data structures: the coordinates and connectivity of vertices that form polygons. In a certain range, one can move the vertices without changing mesh topology or tangling the mesh. The moving polygonal mesh method uses this idea and implements it in an iterative manner.

1. Initialization: Given an initial physical mesh $\mathcal{T}^{(0)}$ for Ω ;
2. Outer iteration ($k = 0, 1, \dots$):
 - (a) Update the metric tensor $\mathbb{M}^{(k)}$ based on the information available at the current iteration. The information includes the current mesh $\mathcal{T}^{(k)}$ and the physical solution $u^{(k)}$ that is obtained by solving the underlying PDE on the current mesh $\mathcal{T}^{(k)}$;
 - (b) Find a way to move the vertices of the physical mesh so that the new mesh $\mathcal{T}^{(k+1)}$ has a better quality under the metric $\mathbb{M}^{(k)}$.

We now give a detailed description of Step 2b. The moving mesh strategy is closely related to $Q_{ali,2}$ and $Q_{eq,2}$ (with piecewise barycentric mappings and subdivision (b)). Recall that they are the toughest set of mesh quality measures among the three, and it is reasonable to expect that the others can be satisfied when they are satisfied. Moreover, $Q_{ali,2}$ and $Q_{eq,2}$ are associated with a triangulation of each polygon in the mesh. One advantage of using piecewise barycentric mappings to define local mappings \mathcal{F}_T is that it allows us to reuse part of the code previously developed in [24] for moving triangular meshes. The triangular subdivisions of all $T \in \mathcal{T}$ used in defining \mathcal{F}_T , combined together, can be viewed as a triangular mesh itself. On the other hand, we should emphasize that *the polygonal moving mesh algorithm is substantially different from a triangular moving mesh algorithm since the former aims to optimize each polygon whereas the latter focuses on the quality of each triangle*. As shown in Fig. 2.5, some of the sub-triangles of a good polygon can indeed have a bad shape.

We assume that a reference computational mesh $\hat{\mathcal{T}}_C$ has been chosen for the mesh movement purpose. For example, we can use a CVT generated by Lloyd’s algorithm. We denote the triangular meshes resulting from the triangulation associated with $Q_{ali,2}$ and $Q_{eq,2}$ for $\hat{\mathcal{T}}_C$ and $\mathcal{T}^{(k+1)}$ by $\mathcal{T}_{\hat{\mathcal{T}}_C}$ and $\mathcal{T}_{\mathcal{T}^{(k+1)}}$, respectively. Consider the function

$$I_h(\{\hat{\mathbf{x}}_i\}, \{\mathbf{x}_i^{(k+1)}\}) = \sum_{K \in \mathcal{T}_{\mathcal{T}^{(k+1)}}} |K| G(J_K, \det(J_K), \mathbb{M}_K), \quad (3.1)$$

where $\{\hat{\xi}_i\}$ and $\{\mathbf{x}_i^{(k+1)}\}$ denote the coordinates of the vertices of $\mathcal{T}_{\hat{\mathcal{T}}_C}$ and $\mathcal{T}_{\mathcal{T}^{(k+1)}}$, respectively, \mathbb{M}_K is the average of $\mathbb{M}^{(k)}$ on K , J_K is the inverse of the Jacobian matrix \mathbb{J}_K of the affine mapping from $K_C \in \mathcal{T}_{\hat{\mathcal{T}}_C}$ to $K \in \mathcal{T}_{\mathcal{T}^{(k+1)}}$, and

$$\begin{aligned} G(J_K, \det(J_K), \mathbb{M}_K) &= \frac{1}{3} \sqrt{\det(\mathbb{M}_K)} (\text{trace}(J_K \mathbb{M}_K^{-1} J_K^T))^2 \\ &\quad + \frac{4}{3} \sqrt{\det(\mathbb{M}_K)} \left(\frac{\det(J_K)}{\sqrt{\det(\mathbb{M}_K)}} \right)^2. \end{aligned} \quad (3.2)$$

The function (3.1) is a discretization of a continuous meshing functional [20, 24] and it is known that minimizing I_h will tend to make the mesh to satisfy the alignment and equidistribution conditions associated with $\mathbb{M}^{(k)}$ and thus to have a better quality under the metric $\mathbb{M}^{(k)}$.

Since $\mathcal{T}_{\hat{\mathcal{T}}_C}$ is known, the new mesh $\{\mathbf{x}_i^{(k+1)}\}$ or $\mathcal{T}_{\mathcal{T}^{(k+1)}}$ can be obtained either by directly minimizing I_h or by first differentiating I_h with respect to $\{\mathbf{x}_i^{(k+1)}\}$ and then solving the corresponding algebraic equations. In either case, an iterative method has to be used because I_h is highly nonlinear. The iterative solution of the new mesh forms the inner iteration.

Recall that $\mathbb{M}^{(k)}$ is defined on $\mathcal{T}^{(k)}$. During the inner iteration, the metric tensor needs to be updated constantly (through interpolation) on approximate meshes of $\mathcal{T}^{(k+1)}$ that will have different vertex locations than $\mathcal{T}^{(k)}$. This can be expensive even if linear interpolation is used.

To avoid this difficulty, an indirect method for finding $\mathcal{T}_{\mathcal{T}^{(k+1)}}$ is used. We replace (3.1) with $I_h(\{\xi_i\}, \{\mathbf{x}_i^{(k)}\})$ (the same form but with difference meshes), where $\{\xi_i\}$ denotes the vertices of a computational mesh $\mathcal{T}_{\mathcal{T}^{(k)}}$ and $\{\mathbf{x}_i^{(k)}\}$ denotes the vertices of the physical mesh $\mathcal{T}_{\mathcal{T}^{(k)}}$. We then minimize the new function with respect to $\{\xi_i\}$ while $\{\mathbf{x}_i^{(k)}\}$ remains fixed. An advantage of this is that, since $\mathcal{T}_{\mathcal{T}^{(k)}}$ is fixed during the inner iteration, there is no need for updating $\mathbb{M}^{(k)}$. Moreover, minimizing $I_h(\{\xi_i\}, \{\mathbf{x}_i^{(k)}\})$ also tends to make the mesh to satisfy the alignment and equidistribution conditions associated with $\mathbb{M}^{(k)}$. Thus, it is reasonable to expect that the correspondence

$$\{\mathbf{x}_i^{(k)}\} = \Phi_h(\{\xi_i\})$$

is close to the correspondence

$$\{\mathbf{x}_i^{(k+1)}\} = \Psi_h(\{\hat{\xi}_i\}),$$

which is determined by minimizing (3.1). Then, we can define the new mesh as

$$\{\mathbf{x}_i^{(k+1)}\} = \Phi_h(\{\hat{\xi}_i\}),$$

which can be computed by linear interpolation as Φ_h is completely determined once $\{\mathbf{x}_i^{(k)}\}$ and $\{\xi_i\}$ are known.

Now, the inner iteration is viewed as to find the steady state of a modified gradient flow of the function $I_h(\{\xi_i\}, \{\mathbf{x}_i^{(k)}\})$. The mesh equation is recorded below without derivation. The interested reader is referred to [24] for detailed derivation.

$$\begin{cases} \frac{d\xi_i}{dt} = \frac{P_i}{\tau} \sum_{K \in \omega_i} |K| \mathbf{v}_{iK}^K, & i = 1, \dots, N_v, \quad t > 0 \\ \xi_i(0) = \hat{\xi}_i, & i = 1, \dots, N_v \end{cases} \quad (3.3)$$

where N_v is the number of vertices of $\mathcal{T}_{\mathcal{T}^{(k)}}$, ω_i is the patch of triangles associated with vertex $\mathbf{x}_i^{(k)}$ in $\mathcal{T}_{\mathcal{T}^{(k)}}$, i_K is the local index of $\mathbf{x}_i^{(k)}$ in K , $\mathbf{v}_{i_K}^K$ is the local mesh velocity associated with the i_K^{th} vertex of K , $\tau > 0$ is a constant parameter used to adjust the time scale of mesh movement, and $P = (P_1, \dots, P_{N_v})$ is a positive function used to make the mesh equation to have desired invariance properties. The local velocities are given by

$$\begin{bmatrix} (\mathbf{v}_1^K)^t \\ (\mathbf{v}_2^K)^t \end{bmatrix} = -E_K^{-1} \frac{\partial G}{\partial J_K} - \frac{\partial G}{\partial \det(J_K)} \frac{\det(E_{K_c})}{\det(E_K)} E_{K_c}^{-1}, \quad \mathbf{v}_0^K = -\sum_{j=1}^2 \mathbf{v}_j^K, \quad (3.4)$$

where

$$\begin{aligned} \frac{\partial G}{\partial J_K} &= \frac{4}{3} \sqrt{\det(\mathbb{M}_K)} \text{trace}(J_K \mathbb{M}_K^{-1} J_K^T) \mathbb{M}_K^{-1} J_K^T, \\ \frac{\partial G}{\partial \det(J_K)} &= \frac{8}{3} \frac{\det(J_K)}{\sqrt{\det(\mathbb{M}_K)}}. \end{aligned}$$

The balancing function in (3.3) is chosen to be $P_i = \det(\mathbb{M}(\mathbf{x}_i^{(k)}))^{\frac{1}{2}}$ such that (3.3) is invariant under the scaling transformation $\mathbb{M} \rightarrow c\mathbb{M}$.

The mesh equation (3.3) should be modified properly for boundary vertices. For fixed boundary vertices, the corresponding equations should be replaced by

$$\frac{d\boldsymbol{\xi}_i}{dt} = 0.$$

For boundary vertices on a curve represented by $\phi(\boldsymbol{\xi}) = 0$, the corresponding mesh equations should be modified such that its normal component along the curve is zero, i.e.,

$$\nabla \phi(\boldsymbol{\xi}_i) \cdot \frac{d\boldsymbol{\xi}_i}{dt} = 0.$$

The mesh equation (3.3), along with proper modification for boundary vertices, can be solved by any ODE solver. (Matlab's ODE solver ode15s is used in our computation.) In principle, it should be integrated until a steady state is obtained. Since finding $\boldsymbol{\xi}_i$ just represents one step of the outer iteration, we integrate (3.3) only up to $t = 1$ (with $\tau = 1/300$) to save CPU time.

We now discuss the choice and computation of the metric tensor. We choose the metric tensor based on optimizing the L^2 norm of error for piecewise linear interpolation [22, 27]. The main reason for this choice is that it is simple, problem independent, and effective. It reads as

$$\mathbb{M} = \det(\alpha_h I + |H(u_h)|)^{-\frac{1}{6}} [\alpha_h I + |H(u_h)|], \quad (3.5)$$

where u_h is an approximate solution, $H(u_h)$ is the recovered Hessian of u_h , $|H(u_h)|$ is the eigen-decomposition of $H(u_h)$ with the eigenvalues being replaced by their absolute values, and the regularization parameter $\alpha_h > 0$ is chosen such that

$$\int_{\Omega} \sqrt{\det(\mathbb{M})} d\mathbf{x} = 2 \int_{\Omega} \det(|H(u_h)|)^{\frac{1}{3}} d\mathbf{x}. \quad (3.6)$$

It has been shown in [28] that when the recovered Hessian satisfies a closeness assumption, a linear finite element solution of an elliptic boundary value problem on

a simplicial mesh computed using the moving mesh algorithm converges at a second order rate as the mesh is refined. Numerical examples presented later show that the same strategy seems to work well for polygonal meshes too. We use a Hessian recovery method based on a least squares fit. More specifically, a quadratic polynomial is constructed locally for each vertex via least squares fitting to neighboring nodal function values and an approximate Hessian at the vertex is then obtained by differentiating the polynomial.

Finally, we point out that currently there is no built-in mechanism in the algorithm to prevent mesh elements from becoming non-convex, or to prevent mesh-tangling. Such issues, as well as the convergence of the algorithm, are generally difficult to analyze theoretically [5]. On the other hand, the continuous meshing functional corresponding to $I_h(\{\xi_i\}, \{\mathbf{x}_i^{(k)}\})$ is coercive and polyconvex and has a minimizer [26]. Moreover, numerical results to be given in Section 4 show that the algorithm is efficient and robust.

4. Numerical results. In this section, we present numerical results obtained with the moving mesh algorithm given in Section 3 for Laplace's equation $-\Delta u = f$ in $\Omega = (0, 1)^2$ subject to the Dirichlet boundary condition. Two examples with the following exact solutions are tested,

Example 1: $u = \tanh(40y - 80x^2) - \tanh(40x - 80y^2);$

Example 2: $u = \sqrt{0.5(r - x)} - 0.25r^2, \quad r = \sqrt{x^2 + y^2}.$

These examples are solved on polygonal meshes using the Wachspress finite element method [46], an H^1 conforming finite element method using the Wachspress barycentric coordinates as basis functions. It is known that for smooth exact solutions and sufficiently fine shape-regular quasi-uniform polygonal meshes, the Wachspress finite element method has the asymptotic convergence order $O(h)$ in H^1 semi-norm and $O(h^2)$ in L^2 norm, where h is the characteristic size of the mesh. If one considers a quasi-uniform polygonal mesh with $N \times N$ polygons, the characteristic mesh size h roughly equals N^{-1} . Consequently, the optimal asymptotic order of the approximation error for the Wachspress finite element method can be expressed into $O(N^{-1})$ in H^1 semi-norm and $O(N^{-2})$ in L^2 norm.

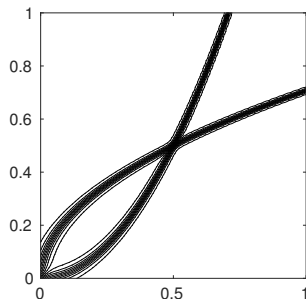


Fig. 4.1: Contour plot of the exact solution for Example 1.

Let us briefly explain the reason why we pick these two examples. For Example 1, the value of u ranges from -2 to 2 in Ω . An equidistant contour plot of u is shown in Fig. 4.1. Apparently, u changes rapidly near the curves $y - 2x^2 = 0$ and $x - 2y^2 = 0$,

while it is almost constant everywhere else. It exhibits a strong anisotropic behavior in the gradient and Hessian near those curves. When a uniform isotropic mesh is used to discretize Example 1, the number of the elements has to be very large to resolve the rapid changes of u near the curves. On the other hand, less elements can be used for the same accuracy when using an adaptive anisotropic mesh, which is dense around the regions with the rapid changes of u and coarse in places where u is almost constant. Here, we will show that the moving mesh algorithm described in Section 3 is capable of generating high quality anisotropic adaptive polygonal meshes optimized for this problem in terms of the L^2 norm of the approximation error.

Example 2 is a well-known example with a corner singularity at $(0,0)$, and the exact solution u is in $H^{\frac{3}{2}-\varepsilon}(\Omega)$ for arbitrarily small $\varepsilon > 0$. When discretized using quasi-uniform meshes, the best approximation error that can be reached has asymptotic order $O(h^{0.5}) = O(N^{-0.5})$ in H^1 semi-norm and $O(h^{1.5}) = O(N^{-1.5})$ in L^2 norm. It is emphasized that, no matter how fine the uniform mesh is, the asymptotic order of approximation error cannot be improved due to the intrinsic low regularity of the exact solution. Later on we will show that adaptive meshes generated by the moving mesh method not only can lead to smaller errors but also improves the convergence order to the optimal $O(N^{-2})$ for the L^2 norm.

For both examples, we set $\Omega_C = \Omega$ and use CVTs generated on Ω_C by Lloyd's algorithm as the reference computational mesh $\hat{\mathcal{T}}_C$. According to the numerical results given in Section 2, the mesh $\hat{\mathcal{T}}_C$ has good quality under the Euclidean metric. We take the initial physical mesh $\mathcal{T}^{(0)} = \hat{\mathcal{T}}_C$.

We start from a reference computational mesh $\hat{\mathcal{T}}_C$ with 32×32 polygons and apply the moving mesh algorithm with 10 outer iterations to Example 1. The initial mesh and the physical meshes after 1 and 10 outer iterations, i.e., $\mathcal{T}^{(0)}$, $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(10)}$, are shown in Fig. 4.2. One can see that the meshes correctly capture the rapid changes of the solution (cf. Fig. 4.1). In addition, a close view of $\mathcal{T}^{(10)}$ near $(0.5, 0.5)$ (shown in Fig. 4.2) clearly shows the anisotropic behavior of the mesh elements. The history of alignment and equidistribution measures is reported in Fig. 4.3. Here, $Q_{ali,1}$, $Q_{ali,2}$ and $Q_{eq,1}$, $Q_{eq,2}$ are computed by comparing the physical mesh with the reference computational mesh $\hat{\mathcal{T}}_C$, while $Q_{ali,3}$ and $Q_{eq,3}$ are computed by comparing each polygon T in the physical mesh with its own reference K_T . One can see that the moving mesh algorithm reduces the mesh quality measures although the reduction is not monotone. Moreover, $Q_{ali,2}$ is slightly bigger than $Q_{ali,1}$ and $Q_{ali,3}$ and $Q_{eq,3}$ is slightly bigger than $Q_{eq,1}$. These are consistent with what we have observed in Section 2. Fig. 4.3 also shows that all three sets of mesh quality measures have very similar evolution patterns.

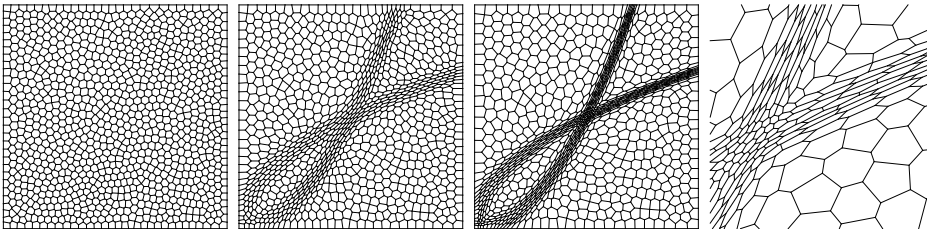


Fig. 4.2: Example 1, mesh with 32×32 cells. From left to right: $\mathcal{T}^{(0)}$, $\mathcal{T}^{(1)}$, $\mathcal{T}^{(10)}$, a close view of $\mathcal{T}^{(10)}$ near $(0.5, 0.5)$.

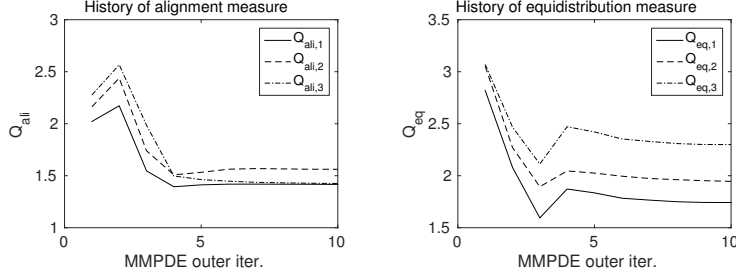


Fig. 4.3: Example 1, mesh with 32×32 cells. History of Q_{ali} and Q_{eq} . Subdivision (b) was used for computing $Q_{ali,2}$ and $Q_{eq,2}$.

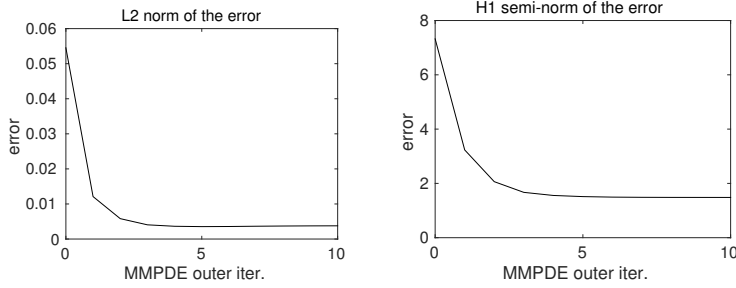


Fig. 4.4: Example 1, a mesh with 32×32 cells. History of L^2 norm and H^1 semi-norm of the error $u - u_h$ is plotted as a function of the outer iteration number.

A more important question is whether these adaptive meshes actually help reduce the approximation error or not. To examine this, we computed the L^2 norm and the H^1 semi-norm of $u - u_h$, where u_h is the finite element solution using the Wachspress finite element method, on these adaptive meshes. The results are reported in Fig. 4.4. The figure clearly shows the effectiveness of the moving mesh method in reducing the approximation error, as both the L^2 norm and the H^1 semi-norm of the error are reduced for about 10 times after applying 10 outer iterations. Interestingly, although the algorithm does not reduce Q_{ali} or Q_{eq} monotonically, it seems to reduce the L^2 norm and the H^1 semi-norm of the error monotonically for Example 1. We also notice that the majority of reduction occurs within the first few outer iterations.

With the above observations, we continue testing the method for Example 1, on meshes with different numbers of cells. Consider meshes with $N \times N$ polygonal cells, for $N = 8, 16, 32, 64$, and 128 . Again, the reference computational meshes are taken as CVTs generated by Lloyd's algorithm. Optimal rates of convergence, $O(N^{-1})$ in H^1 semi-norm and $O(N^{-2})$ in L^2 norm, usually cannot be achieved when the mesh is not fine enough to resolve all detail of the solution. In Table 4.1, it can be seen that on quasi-uniform mesh $\mathcal{T}^{(0)}$, asymptotic order of errors in H^1 semi-norm is less than $O(N^{-1})$ and improves as N increases. The asymptotic order of the error in L^2 norm on $\mathcal{T}^{(0)}$ appears to be larger than $O(N^{-2})$ at the beginning, which is indeed an indication that very coarse meshes cannot fully resolve the detail of the solution and thus result in bad approximations. Again, when N increases, the asymptotic order of the error in L^2 norm improves until it reaches $O(N^{-2})$.

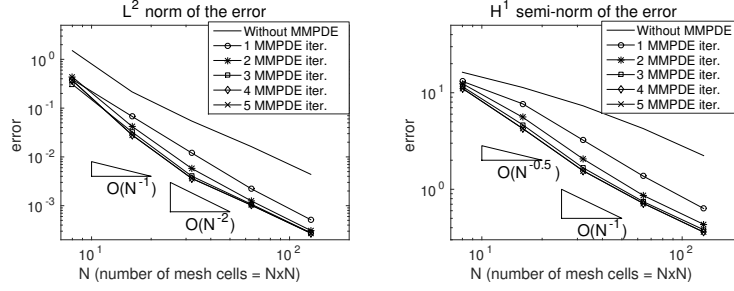


Fig. 4.5: Example 1, mesh with $N \times N$ cells, for $N = 8, 16, 32, 64$, and 128 . The L^2 norm and the H^1 semi-norm of the error $u - u_h$ after different numbers of MMPDE outer iterations are plotted as functions of N .

Table 4.1: Example 1, mesh with $N \times N$ cells, for $N = 8, 16, 32, 64$, and 128 . The L^2 norm and the H^1 semi-norm of the error on $\mathcal{T}^{(0)}$, i.e., no MMPDE iteration, and $\mathcal{T}^{(5)}$, i.e., 5 MMPDE iterations. Here, the asymptotic order of the error is computed using two consecutive meshes with respect to $\frac{1}{N}$.

	On mesh $\mathcal{T}^{(0)}$				On mesh $\mathcal{T}^{(5)}$			
	L^2 norm		H^1 semi-norm		L^2 norm		H^1 semi-norm	
N	error	order	error	order	error	order	error	order
8	1.50e+0		1.63e+1		4.15e-1		1.16e+1	
16	2.16e-1	2.8	1.13e+1	0.5	2.65e-2	4.0	4.17e+0	1.5
32	5.45e-2	2.0	7.32e+0	0.6	3.54e-3	2.9	1.51e+0	1.5
64	1.63e-2	1.7	4.25e+0	0.8	1.01e-3	1.8	7.04e-1	1.1
128	4.39e-3	1.9	2.23e+0	0.9	2.73e-4	1.9	3.51e-1	1.0

Both the H^1 semi-norm and L^2 norm of the approximation error are significantly improved by the anisotropic adaptive mesh algorithm. In Fig. 4.5, the approximation error is reported for different mesh sizes and different numbers of MMPDE outer iterations. We can see that the approximation error can be effectively reduced by performing just a few MMPDE outer iterations. To better compare the quantity of the error, we also list the results on the initial mesh and on the physical mesh after 5 MMPDE outer iterations in Table 4.1. Asymptotic orders in terms of $\frac{1}{N}$ are reported in the table too. Improvements in both the H^1 semi-norm and the L^2 norm and in convergence order can be observed.

Next, we test Example 2 under the same settings as for Example 1. We start from an initial mesh with 32×32 cells and apply the MMPDE algorithm. The initial mesh and physical meshes after 1 and 10 outer iterations of MMPDE are shown in Fig. 4.6. The history of Q_{ali} and Q_{eq} and the L^2 norm and H^1 semi-norm of the error is reported in Figs. 4.7 and 4.10. One may notice that $Q_{ali,1}$, $Q_{ali,2}$, and $Q_{ali,3}$ stay almost constant while $Q_{eq,1}$, $Q_{eq,2}$, and $Q_{eq,3}$ increase slightly during the MMPDE iterations. We first point out that the corner singularity in this example is essentially isotropic. Therefore, any isotropic mesh has good alignment under any metric based on the recovered Hessian of the solution. As a result, we expect that Q_{ali} remains small and constant during the MMPDE outer iterations. To better illustrate

this, in Fig. 4.8 we plot the piecewise constant functions $q_{ali,1}$ and $q_{eq,1}$ defined in Approximation 1, i.e.,

$$q_{ali,1} = \frac{\text{trace}([E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}]^t \mathbb{M}_T [E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}])}{2 \det([E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}]^t \mathbb{M}_T [E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}])^{1/2}},$$

$$q_{eq,1} = \frac{\det(E_T E_{T_C}^t (E_{T_C} E_{T_C}^t)^{-1}) \sqrt{\det(\mathbb{M}_T)}}{\sigma_{h,1}},$$

where $\sigma_{h,1}$ is defined as in (2.15). The ranges of $q_{ali,1}$ and $q_{eq,1}$ are $[1, \infty)$ and $(0, \infty)$, respectively. Notice that $Q_{ali,1}$ and $Q_{eq,1}$ are the maximum norm of $q_{ali,1}$ and $q_{eq,1}$. The left panel of Fig. 4.8 shows that $q_{ali,1}$ stays almost constant on the entire domain and its distribution is independent of the corner singularity. (We plot only $q_{ali,1}$ and $q_{eq,1}$ in Fig. 4.8 since the corresponding quantities for other mesh quality measures behave similarly.)

On the other hand, we recall that the exact Hessian for this example is infinite at $(0, 0)$. Thus, the more adaptive the mesh is, more elements are moved toward the origin and the computed Hessian has greater values there and overall it is becoming less smoother. As a consequence, it is harder to generate a mesh with perfect equidistribution and we see that Q_{eq} is getting bigger as the MMPDE outer iterations. Moreover, Q_{eq} is defined as the maximum norm of q_{eq} , which tells about the mesh quality of worst polygons. As shown in the right panel of Fig. 4.8, worst polygons occur near the origin. The L^2 norm of the quantities is shown in Fig. 4.9. The results show that the L^2 norm decreases almost monotonically, indicating that the MMPDE algorithm has successfully improved the quality of the majority of mesh elements although the worst polygons remain “bad”.

The L^2 norm and H^1 semi-norm of the approximation error are shown in Fig. 4.10. Both of them decrease monotonically for Example 2, which further suggests that the MMPDE algorithm works effectively for Example 2.

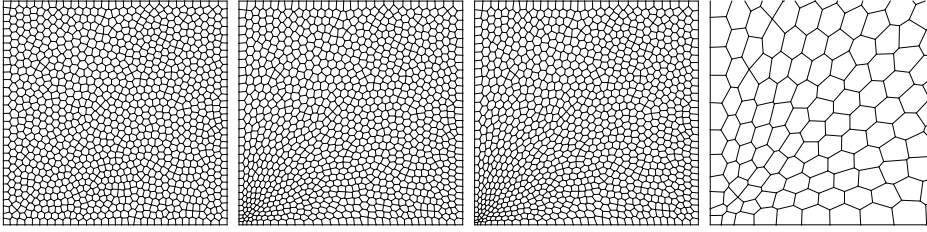


Fig. 4.6: Example 2, mesh with 32×32 cells. From left to right: $\mathcal{T}^{(0)}$, $\mathcal{T}^{(1)}$, $\mathcal{T}^{(10)}$, a close view of $\mathcal{T}^{(10)}$ near the origin.

Finally, we test the MMPDE algorithm for Example 2 on $N \times N$ meshes with different N . As mentioned earlier, the error $u - u_h$ has at best the asymptotic order $O(N^{-0.5})$ in H^1 semi-norm and $O(N^{-1.5})$ in L^2 norm on $N \times N$ quasi-uniform meshes. This can be seen clearly in Table 4.2. In Fig. 4.11 and Table 4.2, the error for Example 2 under different mesh sizes and MMPDE outer iterations is reported. From Fig. 4.11, it is clear that increasing the number of MMPDE outer iterations affects the order of L^2 norm and H^1 semi-norm of the approximation error. The numerical values of the asymptotic order reported in Table 4.2 give a clearer comparison. After 5 MMPDE outer iterations, the L^2 norm of the approximation error achieves almost

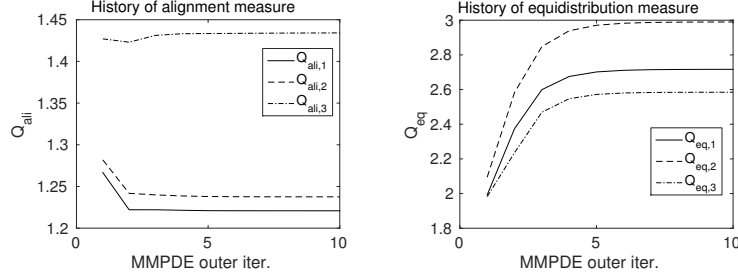


Fig. 4.7: Example 2, mesh with 32×32 cells. History of Q_{ali} and Q_{eq} . Subdivision (b) was used for $Q_{ali,2}$ and $Q_{eq,2}$.

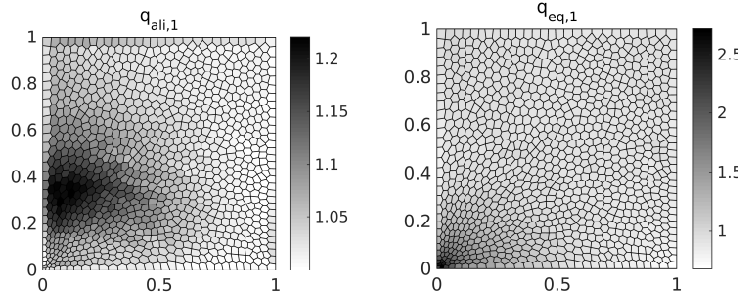


Fig. 4.8: Example 2, distribution of $q_{ali,1}$ and $q_{eq,1}$ on mesh with 32×32 cells after 10 MMPDE outer iterations.

$O(N^{-2})$. The asymptotic order of H^1 semi-norm also improves, but not to the optimal $O(N^{-1})$ order. This may be because the algorithm is designed to optimize the L^2 norm of the error instead of the H^1 semi-norm of the error. To see this, we use the metric tensor

$$\mathbb{M} = \det(\alpha_h I + |H(u_h)|)^{-\frac{1}{4}} \|\alpha_h I + |H(u_h)|\|^{\frac{1}{2}} [\alpha_h I + |H(u_h)|], \quad (4.1)$$

where α_h is determined through the equation (3.6). This metric tensor is based on minimizing the H^1 semi-norm of linear interpolation error [22]. The numerical results are shown in Fig. 4.12 and Table 4.3. The convergence order in the H^1 semi-norm improves (around 0.9) while maintaining the second order rate for the L^2 norm.

5. Conclusions. In the previous sections we have studied anisotropic mesh quality measures and adaptation for polygonal meshes. Three sets of alignment (for shape) and equidistribution (for size) quality measures have been developed. They can be viewed as numerical discretizations or approximations of continuous-level measures (2.10). Their major features are summarized in subsection 2.5. Numerical examples show that they all can give good indicators for the quality of polygonal meshes.

Among the three sets of measures, $Q_{ali,2}$ (2.16) and $Q_{eq,2}$ (2.17) provide the toughest measurement. One of the special cases for defining $Q_{ali,2}$ and $Q_{eq,2}$ is to use piecewise linear generalized barycentric mappings which can be based on triangulation of each polygonal cell into triangles. The collection of those triangles forms a triangular mesh, through which many adaptive mesh algorithms designed for tri-

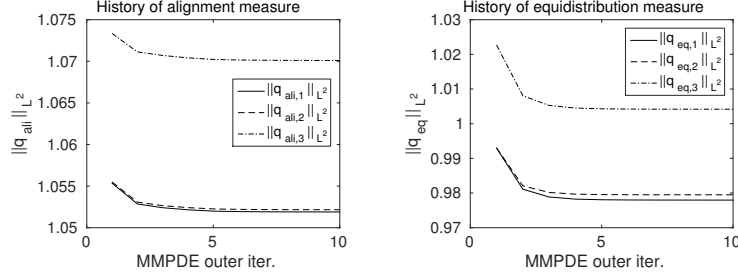


Fig. 4.9: Example 2, mesh with 32×32 cells. History of $\|q_{ali}\|_{L^2}$ and $\|q_{eq}\|_{L^2}$. Sub-division (b) was used for $Q_{ali,2}$ and $Q_{eq,2}$.

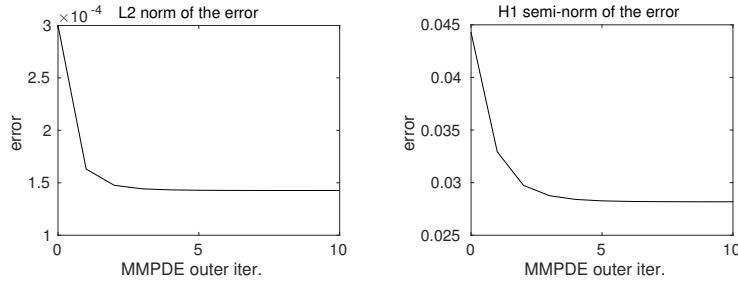


Fig. 4.10: Example 2, mesh with 32×32 cells. History of L^2 norm and H^1 semi-norm of the error $u - u_h$.

angular meshes can be adopted for polygonal mesh adaptation. Though special care needs to be taken since a good polygonal cell can have bad triangles and algorithms should focus on improving the quality of polygonal cells instead of triangles.

Along this line, an anisotropic adaptive polygonal mesh algorithm based on the MMPDE moving mesh method has been devised. For a given reference computational (polygonal) mesh, it moves the mesh vertices such that the adaptive polygonal mesh has a good quality with reference to the reference one. In the numerical examples, the reference computational mesh has been taken as a CVT generated with Lloyd's algorithm. Numerical results confirm that the proposed anisotropic adaptive polygonal mesh algorithm is able to produce desired mesh concentration and lead to a more accurate solution than using a uniform polygonal mesh.

Finally, we would like to point out that the main idea in this work can be generalized to three-dimensions to define anisotropic polyhedral mesh quality measures and generate anisotropic adaptive polyhedral meshes. However, the technical details of the implementation and the effectiveness and robustness of the method in three-dimensions remain to be examined in future work.

REFERENCES

- [1] M. Alexa, D. Cohen-OR, and D. Levin. As-rigid-as-possible shape interpolation. *In Proceedings of SIGGRAPH*. 157–164, New Orleans, 2000.
- [2] T. Apel. *Anisotropic finite elements: local estimates and applications*. B. G. Teubner Stuttgart, Leipzig, 1999. Series of Advances in Numerical Mathematics.

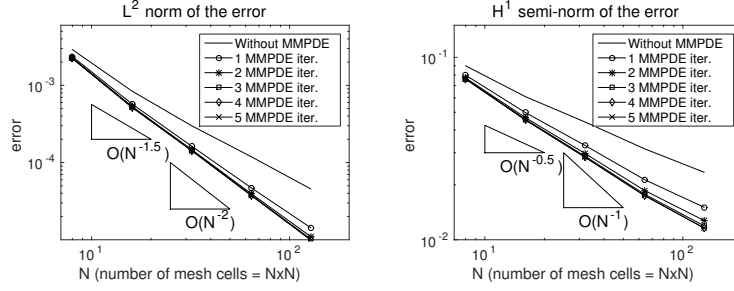


Fig. 4.11: Example 2, mesh with $N \times N$ cells, for $N = 8, 16, 32, 64, 128$. The L^2 norm and H^1 semi-norm of the error $u - u_h$ after different numbers of MMPDE outer iterations are plotted as functions of N .

Table 4.2: Example 2, mesh with $N \times N$ cells, for $N = 8, 16, 32, 64$, and 128 . The L^2 norm and H^1 semi-norm of the error on $\mathcal{T}^{(0)}$, i.e., no MMPDE iteration, and $\mathcal{T}^{(5)}$, i.e., 5 MMPDE iterations. The asymptotic order of the error is computed using two meshes with consecutive number of cells with respect to $\frac{1}{N}$.

	On mesh $\mathcal{T}^{(0)}$				On mesh $\mathcal{T}^{(5)}$			
	L^2 norm		H^1 semi-norm		L^2 norm		H^1 semi-norm	
N	error	order	error	order	error	order	error	order
8	2.90e-3		8.99e-2		2.23e-3		7.59e-2	
16	8.43e-4	1.8	6.09e-2	0.6	5.19e-4	2.1	4.53e-2	0.7
32	3.00e-4	1.5	4.43e-2	0.5	1.43e-4	1.9	2.83e-2	0.7
64	1.19e-4	1.3	3.15e-2	0.5	3.74e-5	1.9	1.74e-2	0.7
128	4.55e-5	1.4	2.35e-2	0.4	9.84e-6	1.9	1.14e-2	0.6

- [3] T. Apel and M. Dobrowolski. Anisotropic interpolation with applications to the finite element method. *Computing*, 47:277–293, 1992.
- [4] F. Brezzi, K. Lipnikov, and M. Shashkov. Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM J. Numer. Anal.*, 43:1872–1896, 2005.
- [5] C.J. Budd, W. Huang, and R.D., Russel. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.
- [6] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *Int. J. Numer. Meth. Fluids*, 25:475–491, 1997.
- [7] L. Chen, P. Sun, and J. Xu. Optimal anisotropic meshes for minimizing interpolation errors in L^p norm. *Math. Comp.*, 76:179–204, 2006.
- [8] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 47:1319–1365, 2009.
- [9] E. F. D’Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Stat. Comput.*, 10:1063–1075, 1989.
- [10] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Rev.*, 41:637–676, 1999.
- [11] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numer. Math.*, 89:641–667, 2001.
- [12] L. Formaggia and S. Perotto. Anisotropic error estimates for elliptic problems. *Numer. Math.*, 94:67–92, 2003.
- [13] M.S. Floater. Mean value coordinates. *Comp. Aided Geom. Design*, 20:19–27, 2003.
- [14] M.S. Floater, K. Hormann, and G. Kós. A general construction of barycentric coordinates over convex polygons. *Advances in Comp. Math.*, 24:311–331, 2006.

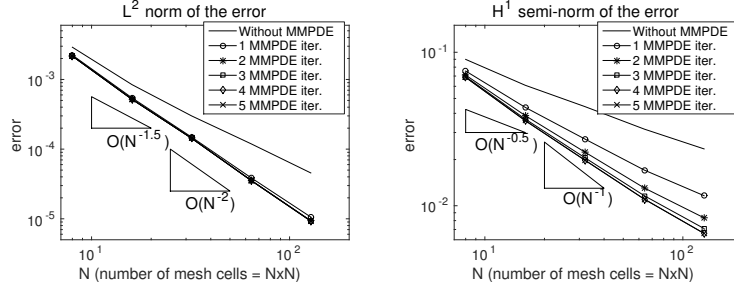


Fig. 4.12: Example 2, mesh with $N \times N$ cells, for $N = 8, 16, 32, 64, 128$. Optimized for H^1 semi-norm instead of for L^2 norm. The L^2 norm and H^1 semi-norm of the error $u - u_h$ after different numbers of MMPDE outer iterations are plotted as functions of N . H^1 semi-norm based metric tensor (4.1) is used.

Table 4.3: Example 2, mesh with $N \times N$ cells, for $N = 8, 16, 32, 64$, and 128. Optimized for H^1 semi-norm instead of for L^2 norm. The L^2 norm and H^1 semi-norm of the error on $\mathcal{T}^{(0)}$, i.e., no MMPDE iteration, and $\mathcal{T}^{(5)}$, i.e., 5 MMPDE iterations. The asymptotic order of the error is computed using two meshes with consecutive number of cells with respect to $\frac{1}{N}$. H^1 semi-norm based metric tensor (4.1) is used.

	On mesh $\mathcal{T}^{(0)}$				On mesh $\mathcal{T}^{(5)}$			
	L^2 norm		H^1 semi-norm		L^2 norm		H^1 semi-norm	
N	error	order	error	order	error	order	error	order
8	2.90e-03		8.99e-02		2.17e-03		6.83e-02	
16	8.43e-04	1.8	6.09e-02	0.6	5.18e-04	2.1	3.55e-02	0.9
32	3.00e-04	1.5	4.43e-02	0.5	1.43e-04	1.9	1.95e-02	0.9
64	1.19e-04	1.3	3.15e-02	0.5	3.55e-05	2.0	1.08e-02	0.9
128	4.55e-05	1.4	2.35e-02	0.4	9.28e-06	1.9	6.36e-03	0.8

- [15] M.S. Floater. Generalized barycentric coordinates and applications. *Acta Numerica*, 24:161–214, 2015.
- [16] M.S. Floater and J. Kosinka. On the injectivity of Wachspress and mean value mappings between convex polygons. *Adv. Comput. Math.*, 32:163–174, 2010.
- [17] G.J. Gassner, F. L rcher, C.-D. Munza, and J.S. Hesthaven. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *J. Comp. Phys.*, 228:1573–1590, 2009.
- [18] A. Gillette, A. Rand, and C. Bajaj. Error estimates for generalized barycentric coordinates. *Adv. Comput. Math.*, 37:417–439, 2012.
- [19] R. Herbin. Finite volume methods for diffusion convection equations on general meshes. in *Finite volumes for complex applications, Problems and Perspectives*, 153–160, Hermes, 1996.
- [20] W. Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174:903–924, 2001.
- [21] W. Huang. Measuring mesh qualities and application to variational mesh adaptation. *SIAM J. Sci. Comput.*, 26:1643–1666, 2005.
- [22] W. Huang. Metric tensors for anisotropic mesh generation. *J. Comput. Phys.*, 204:633–665, 2005.
- [23] W. Huang. Mathematical principles of anisotropic mesh adaptation. *Comm. Comput. Phys.*, 1:276–310, 2006.

- [24] W. Huang and L. Kamenski. A geometric discretization and a simple implementation for variational mesh generation and adaptation. (*submitted*), 2014. (arXiv:1410.7872).
- [25] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle. *SIAM J. Numer. Anal.*, 31:709–730, 1994.
- [26] W. Huang and R. D. Russell. *Adaptive Moving Mesh Methods*. Springer, New York, 2011. Applied Mathematical Sciences Series, Vol. 174.
- [27] W. Huang and W. Sun. Variational mesh adaptation II: error estimates and monitor functions. *J. Comput. Phys.*, 184:619–648, 2003.
- [28] L. Kamenski and W. Huang. How a nonconvergent recovered Hessian works in mesh adaptation. *SIAM J. Numer. Anal.*, 52:1692–1708, 2014. (arXiv:1211.2877).
- [29] Y. Lipman. Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH*, 31, Article 108, 13 pages, 2012.
- [30] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. *J. Comp. Phys.*, 257:1163–1227, 2014.
- [31] A. Liu and B. Joe. Relationship between tetrahedron quality measures. *BIT*, 34:268–287, 1994.
- [32] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28:129–137, 1982.
- [33] G. Manzini, A. Russo and N. Sukumar. New perspectives on polygonal and polyhedral finite element methods. *Math. Models Methods Appl. Sci.*, 24:1665–1699, 2014.
- [34] M. Meyer, A. Barr, H. Lee and M. Desbrun. Generalized barycentric coordinates for irregular polygons. *J. Graph. Tools*, 7:13–22, 2002.
- [35] L. Mu, X. Wang, and Y. Wang. Shape regularity conditions for polygonal/polyhedral meshes, exemplified in a discontinuous Galerkin discretization. *Numer. Meth. Part. Diff. Eq.*, 31:308–325, 2015.
- [36] L. Mu, J. Wang, and X. Ye. Weak Galerkin finite element methods on polytopal meshes. *Inter. J. Numer. Anal. Model.*, 12:31–53, 2015.
- [37] M. Picasso. An anisotropic error indicator based on zienkiewicz-zhu error estimator: Application to elliptic and parabolic problems. *SIAM J. Sci. Comput.*, 24:1328–1355, 2003.
- [38] T. Schneider, K. Hormann, and M.S. Floater. Bijective composite mean value mappings. *SGP '13 Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, 32:137–146, 2013.
- [39] J. R. Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *Proceedings, 11th International Meshing Roundtable*, pages 115–126, Sandia National Laboratories, Albuquerque, NM, 2002.
- [40] N. Sukumar and A. Tabarraei. Conforming polygonal finite elements. *Int. J. Numer. Meth. Engrg.*, 61:2045–2066, 2004.
- [41] N. Sukumar and E.A., Malsch. Recent advances in the construction of polygonal finite element interpolants. *Arch. Comput. Methods Engrg.*, 13:129–163, 2006.
- [42] W. Sun, J. Wu, and X. Zhang. A family of linearity-preserving schemes for anisotropic diffusion problems on arbitrary polyhedral grids. *Comput. Methods Appl. Mech. Engrg.*, 267:418–433, 2013.
- [43] C. Talischi, G.H. Paulino, and C. Le. Honeycomb Wachspress finite elements for structural topology optimization. *Struct. Multidis. Opt.*, 37:569–583, 2009.
- [44] C. Talischi, G.H. Paulino, A. Pereira, and I.F.M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *I. J. Numer. Meth. Engrg.*, 82:671–698, 2010.
- [45] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L.D. Marini, and A. Russo. Basic principles of virtual element methods. *Math. Models Meth. Appl. Sci.*, 23:199–214, 2013.
- [46] E.L. Wachspress. *A rational finite element basis*. Academic Press, New York, 1975.
- [47] J. Wang and X. Ye. A Weak Galerkin mixed finite element method for second-order elliptic problems. *Math. Comp.*, 83:2101–2126, 2014.
- [48] M. Wicke, M. Botsch and M. Gross. A finite element method on convex polyhedra. *Comp. Graph. Forum, Proceedings Eurographics*, 26(3):255–364, 2007.
- [49] G.M. Ziegler. *Lectures on polytopes*. Springer, New York, 1995.